

# **1 Einleitung**

## **1.1 Motivation**

## **1.2 Aufgabenstellung**

## **1.3 Aufbau der Arbeit**

## 2 Grundlagen

Der folgende Abschnitt gibt einen kurzen Überblick über BPMN Diagramme und erläutert mögliche Ansätze des Extrahierens von BPMN-Diagrammen aus FLOSS-GitHub-Projekten, die von uns ausprobiert wurden.

### 2.1 BPMN-Diagramme

Die Business Process Model and Notation (BPMN, deutsch Geschäftsprozessmodell und -notation) ist ein öffentlicher Standard, der von OMG und BPMI verwaltet wird. Das Hauptziel von BPMN ist es, eine Notation bereitzustellen, die für alle Geschäftsbenutzer leicht verständlich ist, von den Business-Analysten, die die ersten Entwürfe der Prozesse erstellen, bis zu den technischen Entwicklern, die für die Implementierung der Technologie verantwortlich sind, die diese Prozesse durchführt, und schließlich auf die Geschäftsleute, die diese Prozesse verwalten und überwachen. Somit schafft BPMN eine standardisierte Brücke für die Lücke zwischen Geschäftsprozessdesign und Prozessimplementierung [5].

BPMN basiert auf der Überarbeitung anderer Notationen und Methoden, insbesondere des Aktivitätsdiagramms der Unified Modeling Language (UML), des UML-EDOC-Geschäftsprozesses, IDEF, ebXML BPSS, des Aktivitäts-Entscheidungsflussdiagramms, RosettaNet, LOVeM und Event Driven Process Chains [4]. Die aktuelle Version von BPMN ist 2.0. Sie definiert unter anderem das XML-basierte Format für Speichern und Übertragung von BPMN Diagrammen.

BPMN verfügt über fünf Kernelementkategorien um Geschäftsprozessdia-

gramme zu bilden. Diese sind:

1. Ablaufobjekte, die die wichtigsten grafischen Elemente von BPMN sind und das Verhalten eines Prozesses definieren. Es gibt drei verschiedene Arten von Flussobjekten:

- Ereignis (als Kreis dargestellt)
- Aktivität (dargestellt als Rechteck mit abgerundeten Ecken), die weiter in Aufgaben und Unterprozesse unterteilt werden kann
- Gateway (angezeigt als Raute)

2. Daten oder Informationen, die entweder innerhalb eines Prozesses verarbeitet oder zwischen verschiedenen Prozessen ausgetauscht werden. Dies umfasst die folgenden fünf BPMN-Elemente:

- Datenobjekt
- Dateneingabe
- Datenausgabe
- Datenspeicher
- Nachricht

3. Verbundene Objekte, die es einer einzelnen Person ermöglichen, Bewegungsobjekte miteinander zu verbinden oder mit Zusatzinformationen zu verbinden. Es gibt drei verschiedene Arten von Verbindungsobjekten:

- Sequenzablauf
- Nachrichtenfluss
- Assoziation

4. Schwimmbahnen, die zur Gruppierung der zuvor genannten primären Modellierungselemente verwendet werden. Es gibt zwei Arten von Gruppierungen in BPMN:

- Schwimmbäder
- (Schwimmen) Bahnen

5. Artefakte, die verwendet werden, um zusätzliche Informationen zum Pro-

zess bereitzustellen. Es gibt zwei Arten von Artefakten:

- Gruppe
- Textanmerkung [4]

## 2.2 Extrahieren von BPMN-Diagrammen aus FLOSS-GitHub-Projekten

Im Vorbereitungsphase unserer Forschung wurden verschiedene Ansetze für die Suche nach GitHub-Projekte, die BPMN Diagramme enthalten, analysiert. Grundlage dafür lieferten uns Publikationen, in denen es um die Suche nach UML-Diagramme enthaltende GitHub-Projekte ging. Siehe: [2], [7] und [6].

Die existierende Ansätze für die Suche nach GitHub-Projekte können in zwei Kategorie unterteilt werden, über die es unter weiter gehen wird. Aber um sie anwenden zu können, wurde zuerst eine statische, zuverlässige und aktualisierte Quelle der GitHub Plattform gebraucht. Diese wird vom GHTorrent-Projekt zur Verfügung gestellt, das Daten für alle öffentlichen und verfügbaren GitHub Projekte sammelt [1]. Es wurde zuerts die GHTorrent-Datensammlung von 01.11.2018 heruntergeladen und mithilfe eines sql-Befehls eine Liste mit Nahmen von allen GitHub Projekten, die nicht gelöscht und nicht abgezweigt wurden, in einer neu-erzeugten Datenbank gespeichert. Es gab insgesamt 61 632 173 Namen der GitHub-Projekte.

### 2.2.1 Herunterladen und Durchsuchen

Nachdem uns eine Liste von Namen aller GitHub Projekte zur Verfügung stand, wurde ein Python Programm für die Suche nach Projekten mit BPMN Diagrammen geschrieben. In diesem Programm wurde dann in einer Schleife über alle Namen der GitHub-Projekte iteriert und bei jeder Iteration mit `git clone` ein Projekt heruntergeladen und durchsucht. Wegen der länger Zeit, die das Herunterladen und Durchsuchen in Anspruch nahmen, war dieses Verfahren für unsere Forschung zu langsam.

### 2.2.2 GitHub API

Ein weiterer Ansatz war die Anwendungsprogrammierschnittstelle (englisch: Application Programming Interface, kurz API) von GitHub zu benutzen, die fast alle benutzerschnittstelle-zugreifbare Funktionen in einer äquivalenten API-Form zur Verfügung stellt [3]. Für jede Anfrage an GitHub API wird ein Ergebnis in JavaScript Objekt Notation (JSON) Format geliefert.

Das Benutzen der GitHub API war wiederum in zwei Varianten möglich:

- Die Such-API zu benutzen, die bei der Suche nach dem gewünschten Artikel hilft. Es ist also möglich beispielsweise einen Benutzer oder wie in unserem Fall eine bestimmte Datei in einem Projekt zu finden. Es wurde wiederum ein Python Programm geschrieben, wo in einer Schleife über allen Namen der GitHub Projekte iteriert wurde. Bei jeder Iteration wurde eine Suchanfrage über API an GitHub geschickt. Zum Beispiel für die suche nach `exampleString`, Benutzer: `exampleUser` und GitHub Projekt: `exampleProject` sah die Anfrage wie folgt aus: `https://api.github.com/search/code?q=exampleString+in:file+user:exampleUser+repo:exampleProject`.

# Literaturverzeichnis

- [1] Georgios Gousios. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13*, pages 233–236, Piscataway, NJ, USA, 2013. IEEE Press.
- [2] Regina Hebig, Truong Ho Quang, Michel R. V. Chaudron, Gregorio Robles, and Miguel Angel Fernandez. The quest for open source projects that use uml: Mining github. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, MODELS '16*, pages 173–183, New York, NY, USA, 2016. ACM.
- [3] Loeliger Jon and McCullough Matthew. *Version Control with Git - Powerful Tools and Techniques for Collaborative Software Development*. Ö'Reilly Media, Inc.", Sebastopol, 2012.
- [4] Vivek Kale. *Enterprise Process Management Systems - Engineering Process-Centric Enterprise Systems using BPMN 2.0*. CRC Press, Boca Raton, Fla, 2018.
- [5] OMG. *Business Process Model and Notation (BPMN)*. Needham USA(2013), 2013.
- [6] Gregorio Robles, Truong Ho-Quang, Regina Hebig, Michel Chaudron, and Miguel Fernández. An extensive dataset of uml models in github. pages 519–522, 05 2017.
- [7] Gregorio Robles, Truong Ho-Quang, Regina Hebig, Michel R. V. Chaudron, and Miguel Angel Fernandez. An extensive dataset of uml models

in github. In *Proceedings of the 14th International Conference on Mining Software Repositories*, MSR '17, pages 519–522, Piscataway, NJ, USA, 2017. IEEE Press.